

PROGRAMMING ASSIGNMENT 2**Due date:** 02.11.2018 10:00 am

Write a C program that implements a “task queue”.

Your program begins by generating tasks. A task is either an insert, a delete, or a search a value in a *sorted list (in ascending order)* with *no duplicates*. Your program assigns a `task_num` starting from 0 and incremented by 1 for each task generation, a `task_type` which is insert, delete, or search task in the list, and a `value` for the list operation, randomly. Each time it generates a new task, it adds it to a task queue. After the completion of task generation, your program pulls the tasks from the queue and processes them one by one, by inserting, deleting, or searching from the list. Your program should have the following data structures and C functions:

```
/* Struct for list nodes */
struct lst_node_s {
    int data;
    struct lst_node_s* next;
};

/* Struct for task nodes */
struct tsk_node_s {
    int task_num; //starting from 0
    int task_type; // insert:0, delete:1, search:2
    int value;
    struct tsk_node_s* next;
};

/* List operations */
int Insert(int value);
int Search(int value);
int Delete(int value);

/* Task queue functions */
void Task_queue(int n); //generate n random tasks for the task queue
void Task_enqueue(int task_num, int task_type, int value); //insert a new task into task queue
int Task_dequeue(int* task_num_p, int* task_type_p, int* value_p); //take a task from task queue
```

Your program will get the number of tasks from the user, and display the task generation and completion information. In the final phase, it should print the contents of the list.

Example execution:

```
./queue 10
Generated 10 random list tasks...
task 0-insert 3: 3 is inserted
task 1-insert 17: 17 is inserted
task 2-delete 6: 6 cannot be deleted
task 3-insert 9: 9 is inserted
task 4-insert 13: 13 is inserted
task 5-search 17: 17 is found
task 6-insert 3: 3 cannot be inserted
```

task 7-insert 0: 0 is inserted
task 8-delete 9: 9 is deleted
task 9-search 5: 5 is not found
Final list:
0 3 13 17

Notes:

- You are required to submit a report that includes explanations about implementations with screenshots of your sample executions.
- You need to work individually, no group work is allowed.
- No late homework will be accepted.
- You can use your own work about linked lists performed in the Lab 3 exercise.

Submission: You are required to submit your **commented** source code and report to CMS. Please create a compressed file including all source files and report; and name it as yourstudentnumber_P2.zip (e.g. If your student number is 201812345678, the file name must be 201812345678_P2.zip).